Automatic Video Matting through Scribble Propagation

Bhoomika Sonane
IIT Gandhinagar
bhoomika.sonane@iitgn.ac.in

Sainandan Ramakrishnan
VJTI
saiyanlife415@gmail.com

Shanmuganathan Raman
IIT Gandhinagar
shanmuga@iitgn.ac.in

ABSTRACT

Video matting is an extension of image matting and is used to extract the foreground matte from an arbitrary background of every frame in a video sequence. An automatic scribbling approach based on the relative motion of the foreground object with respect to the background in a video is introduced for video matting. The proposed scribble propagation and the subsequent isolation of foreground and background is much more intuitive than the conventional trimap propagation approach used for video matting. Alpha maps are propagated according to the optical flow estimated from the consecutive frames to get a preliminary estimate of the foreground and background in the following frame. Accurate scribbles are placed near the boundary of the foreground region for refining the scribbled image with the help of morphological operations. We show that a high quality matte of foreground object can be obtained using a state-of-the-art image matting technique. We show that the results obtained using the proposed method are accurate and comparable with that of other state-of-the-art video matting techniques.

CCS Concepts

•**Computing methodologies** → **Tracking;** *Video segmentation;*

Keywords

Video matting, Optical flow, Computational photography

1. INTRODUCTION

Scribbles are used in the matting problem in order to find a solution for the under-constrained matting equations. The scribbles indicate the regions which can be considered definitely as background (black scribbles) and definitely as foreground (white scribbles). The closed form solution to the natural image matting problem can propagate these scribbles according to a cost function derived by assuming continuity in foreground and background colors in a small neighbourhood around individual pixels and optimal solution can be obtained [17]. This method is closely related to the colorization approach where scribbles of different colors are propagated based on a quadratic cost function [16]. In the case of matting, the function has been modified in order to meet the desired objective.

Another way to roughly segment the foreground and background in a scene is to use trimaps, i.e. dividing the image into three regions - background (black), foreground (white), and the unknown region (gray). This is usually done manually but recently it has also been automatized by employing different binary segmentation algorithms [19]. Trimap, however, is not a reliable choice as it involves a significant amount of user effort in specifying it accurately. Even though trimap reduces the dimensionality of the problem by reducing the number of unknowns to be computed and increases the number of known background and foreground pixels, scribbles are considered to be the better option and user friendly. This is due to the fact that merely with a few strokes of black and white scribbles and a good optimization algorithm, a high quality matte can be estimated.

We model an image I to be a convex combination of foreground F and background B as given by equation 1.

$$I(x,y) = \alpha(x,y) \times F(x,y) + (1-\alpha(x,y)) \times B(x,y) \quad (1)$$

where $\alpha(x,y)$ is the alpha value we obtain from image matting for the pixel (x,y) in the image, ranging in [0,1]. For the definite foreground pixels, the value of α is chosen to be 1, whereas for the definite background, the value of α is chosen to be 0. The pixels with the values between 0 and 1 are termed as mixed pixels. These are the pixels where the foreground and background regions combine indicating partial foreground region coverage.

Video matting problem can be considered as a combination of image matting and tracking problem performed on every frame. To solve the image matting problem, scribbles are needed to reduce the dimensionality of unknown data space. Thus black and white scribbles are propagated through all the frames using optical flow, while generating sufficient scribbles in regions where there are insufficient scribbles present.

The primary contributions of this paper are listed below:

1. Scribbles are drawn only for the initial frame to pull a matte and the same scribbles are propagated along the subsequent frames to get the corresponding alpha matte for every frame by tracking the motion of the objects.

ICVGIP, December 18-22, 2016, Guwahati, India
© 2016 ACM. ISBN 978-1-4503-4753-2/16/12…$15.00
DOI: http://dx.doi.org/10.1145/3009977.3009979

2. Optical flow is integrated to get the motion flow estimation of every pixel and the scribbles are accordingly moved.

3. Due to the motion and probable rotation involved in the scene, the scribbles get reduced and the need to add more scribbles is imminent. To tackle this, scribbles for both the regions are automatically generated using the Bezier curves.

4. We avoid any false marking of scribbles in each frame through local and global morphological operations.

The paper is organised as follows. Section 2 gives the overview of the related work in the area of image and video matting. Section 3 covers the proposed methodology. Section 4 highlights the performance parameters used in the proposed approach for a standard dataset. Section 5 discusses the results along with comparison of relevant methods with the proposed method. The paper is concluded in section 6 with a discussion about how it can be improved in future.

2. RELATED WORK

The matting problem has been a topic of interest in the research related to the compositing of images to some arbitrary backgrounds since the introduction of blue-screen matting which gave the field the impetus that it needed [25]. The matte is computed using a known background in the compositing equation which decreases the number of unknowns in the under-constrained equation. It does not give satisfactory results for the case when foreground has any trace of same color that is being used for background. It is then extended to use two different background colors and thus solving the matting equation. The solution to the under-constrained problem in images is also obtained through Bayesian approach where both the background and the foreground Gaussian distributions are computed using nearby pixel information obtained by the sliding window technique [10]. The method was observed to improve by using mixture of Gaussians instead. These probability distributions are further used to formulate a Bayesian framework to compute required parameters which is solved by *maximum a posteriori* (MAP) technique. One such blue screen technique was given by Mishima, which uses triangular meshes to compute the *alpha* matte [20]. When matte from Mishima method and Bayesian approach are computed and compared with the ground truth, Bayesian approach was observed not to produce *blue-spill* or background spill around the boundary whereas the method introduced by Mishima did. While blue-screen matting and difference matting solves the under-constrained problem, they are fully dependent on the availability of known and more preferably constant coloured background [21]. With an arbitrary background, the matting problem is labeled as natural image matting [20]. For natural matting, knockout and the Ruzon and Tomasi algorithms were proposed and shown to produce artifacts which Bayesian did not produce[6, 7, 23].

The Bayesian approach was further extended along with the use of optical flow and background estimation to be used for video matting [9]. Over the years, the Bayesian approach is adopted in several methods giving substantial results and method involving learnt priors from training sequences to be used for video matting [3]. Other significant growth in the research area of video matting was seen from the start of 21st century. Some novel methods and developments in the field were made over the initial years. One such novel method was to use a set of defocused images that share a common point of view giving *data-rich imaging* approach to video matting [19]. The approach gave a fully automated trimap, without any user assistance by using high frequency texture present in the scene. The matte in this method is computed individually for each frame for synthetic as well as real scenes. Also, since it uses a multi-parameter video camera, the errors occur due to parallax between sensors, under/over exposure and the error due to the amount of blur used in sequences, and also the depth discontinuity.

Some methods segment the different layers, foreground and background in most cases, by commonly used image segmentation algorithms such as Graph-Cut [18, 28]. Multiple frames are taken to be used for segmentation of foreground. Coherent matting proves to be more efficient than Bayesian in cases where tracking has to be used and a more clear boundary is required [18]. Another way to produce good results by handling both texture and non-texture regions is to take control of the merits of both the motion segmentation and alpha matting algorithms [28].

A number of methods have been employed to propagate alpha values of initial frame to subsequent frames to produce a high quality composition of the specific foreground with an arbitrary background [24, 27]. Most of these methods aim to reduce the user interaction to generate trimap for the matting and to generate the most accurate trimap for the current frame. Optical flow is the basic choice to propagate either scribbles or for segmenting the rough boundary between different layers to get an approximate estimation of the trimap. One such state-of-the-art method tracks actual object by a discriminative mode [14]. In the model, the intermediate mattes are used to estimate the location of the object by tracing a temporary contour. The method automatically updates a palette of discriminative colors which define the background and foreground regions of the scene uniquely. The palette is calculated in every frame by studying the color histogram of the two regions and estimating the likelihood of each of the two histograms with respect to each color bin. The approach is used to tackle three main problems - the drift in the matte as in the case of fast moving objects, excessive deformation, and the occlusion problem in the matte. Other similar approaches include [4, 11, 15, 22, 29]. The major limitation of these methods can be observed in complex moving backgrounds and the similarity of color in background and foreground is high, leading to misclassification. A keyframe based scribble propagation method has recently been proposed for HDR tone mapping and video re-coloring applications [12].

3. PROPOSED METHOD

3.1 Overview

To pull a good foreground matte from a scene, scribbles or accurate trimaps are needed in the matting algorithms. Marking the scribbles for each frame to generate matte in a video is a tedious task. For this reason, scribbles are marked just on the first frame of the video and then they are propagated through the successive frames. In our approach, we kept the background(black) scribbles stationary for all the successive frames, i.e., the locations of scribbles for the background pixels are fixed. However, because of the movement of the foreground object in the frame, the background scribbles whose space is taken over by the moving foreground are removed. These locations are carefully replaced with white scribbles to indicate the overlapping of the foreground object on the previously occupied background pixels. To estimate the motion flow of the foreground, a state-of-the-art optical flow algorithm is employed [26]. The white scribbles present in the foreground object are moved in the scene according to the flow field displacement vectors given by the optical flow algorithm. To refine the scribbled image in each frame, morphological operations such as erosion and dilation are employed inside and around the boundary of the propagated alpha maps. Additional techniques such as super-pixel over-segmentation and Bezier curves are used to automatically generate and accurately place the required scribbles and cover most of the heterogeneous regions with sufficient scribbles, in order to extract a good quality matte for every frame.

3.2 Methodology

The proposed video matting algorithm is initialised with two images: one being the first frame of the video and the other image being the same frame with scribbles marked in respective regions of the image to indicate definite foreground and background pixels.

This matte will be propagated to the next frame using optical flow values generated with respect to the current frame and the next frame of the sequence [26]. This is shown in figure 1(c), which is the propagated alpha map of frame 102 of the *Dmitriy* benchmark sequence (figure 1(a), figure 1(b)) generated mid-way during the execution of the algorithm on the full sequence [13]. The next frame will now have an approximate foreground matte, the accuracy of which is not a matter of concern as it will be used only for further processing of newly generated scribbled image. This leads to the refinement of the initial result. The previously generated image with scribbles will propagate its scribbles to the next frame. Considering that the foreground scribbles are the only points to get new locations while tracking the movement of the object of interest and the objects moving in the background are not important to track, we only move the white scribbles according to the optical flow vectors and keep the black scribbles stationary.

Let I_i be the previous frame and I_{i+1} be the current frame. Similarly, let S_i be the scribbled image corresponding to the previous frame and S_{i+1} be the scribbled image to be generated corresponding to the current frame. Let α_i be the alpha map of the previous frame solved using the closed form image matting [17]. The alpha map of the current frame α_{i+1} can be computed from α_i using the equation 2.

$$\alpha_{i+1}(x, y) = \alpha_i(x - d_x, y - d_y) \tag{2}$$

where $d = [d_x, d_y]$ is the displacement vector computed by the optical flow algorithm corresponding to the two frames being considered [26].

The scribbled image corresponding to the current frame (figure 1(b)) is basically the original image (figure 1(a)) of the current frame with the black scribbles (stationary) and white scribbles (propagated) marked on the appropriate locations using the alpha map of the previous frame (frame 101). Equation 3 represents the same.

$$S_{i+1}(x, y) = \begin{cases} 0, & S_i(x, y) = 0 \\ 1, & S_i(x - d_x, y - d_y) = 1 \\ I_{i+1}(x, y), & \text{Otherwise} \end{cases} \tag{3}$$

This is employed in order to guide the placement of background and foreground scribbles on frame 102. Since only the white scribbles are moving with the foreground object, at some point, the foreground will get penetrated by the black scribbles which are kept unmoved at their locations from the previous frame. In our example, this happens at frame 112 (figure 1(d)). This should be avoided. Scribbles of neither color should be present at random points, especially near the boundary region, in order to get the resultant scribbled image devoid of any contradictions. The feedback knowledge from the alpha map of the foreground region propagated from the previous frame (frame 111) will be useful in achieving that. For the foreground object, black scribbles should be near but not too close to its boundary. To employ this heuristic, thresholding the the approximated foreground matte and dilating (figure 1(f)) the resulting mask, α^d_{i+1}, will undergo binary multiplication with the mask generated with only the black scribbles (figure 1(e)), S^b_i. The dilation in the frames is represented as D and erosion as E.

$$\alpha^d_{i+1} = D(\alpha_{i+1}) \tag{4}$$

$$S^b_i = ((I_i - S_i) > 0) \tag{5}$$

$$S^b_i(x, y) \wedge \alpha^d_i(x, y) = 0 \tag{6}$$

Therefore, the resulting mask after multiplication (figure 1(h)) will have white region on the locations where the excess black scribbles penetrate the dilated foreground mask, α^d_{i+1}, and the values elsewhere will be zero. The locations where the binary multiplication gives true values will be taken as intersecting or 'too close to the boundary' scribbled pixels and thereby will not be included in the final background scribble mask for the current frame (figure 1(i)).

With the accumulated error in optical flow and the rotation involved in the foreground object, the white scribbles will eventually be pushed out of the foreground boundary thus preventing crisp segmentation. To avoid this, the same heuristic is employed for the white scribbles by binary multiplying the complement of the eroded mask, α^e_{i+1}, in figure 1(g) with the foreground scribble mask, S^w_i.

$$\alpha^e_{i+1} = E(\alpha_{i+1}) \tag{7}$$

$$S^w_i = ((S_i - I_i) > 0) \tag{8}$$

$$S^w_i(x, y) \wedge \alpha^e_i(x, y) = 1 \tag{9}$$

Hence, by taking the dual of the first heuristic, we are able to prevent the foreground pixels within the bulk of the object of interest from being inaccurately 'spilled over' into the background. Figure 1(j) represents the final image with scribbles along with the further processing that will be described later. The zoomed part of (figure 1(d)) highlighting the penetrating background and overflowing foreground scribbles is shown in figure 2(b) and figure 2(c).

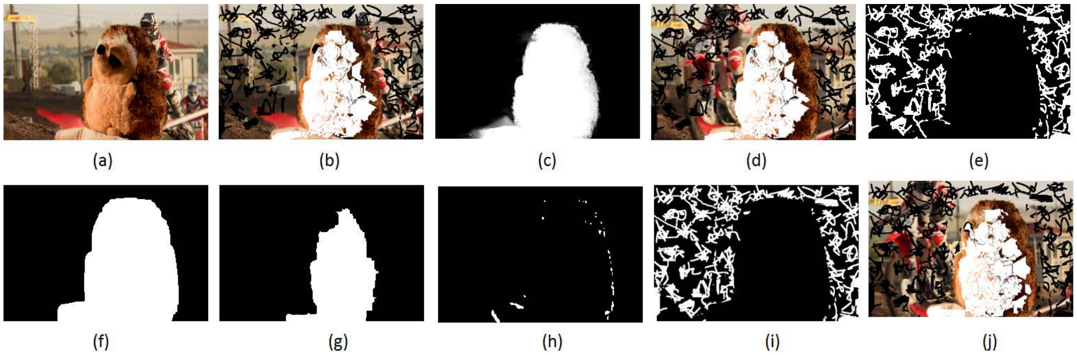

Figure 1: (a) Frame 102, (b) Scribbled 102, (c) Matte 102, (d) Scribbled 112 (Without Morphology), (e) Background scribble mask of (d), (f) Dilated mask of frame 111, (g) Eroded mask of frame 111, (h) Background excess scribbles, (i) Refined Background scribble mask of (d), and (j)Actual Scribbled 112.

Hence,

$$S_{i+1}(x,y) = S_i(x,y) \tag{10}$$

for $(x,y) \in (X,Y)$ where (X,Y) are the locations for which the equations 6 and 9 are satisfied.

With the morphological operations, the amount of black and white scribbles will get reduced and there will eventually be a need to add more scribbles. Usually, in numerous methods, this is achieved by calculating key frames and adding fresh new scribbles to aid the matting process of a video sequence from a fresh perspective. This process is automated in the proposed method with the use of Bezier curves and super-pixel over-segmentation. The image is over-segmented into approximately 50 - 100 super-pixels to group image pixels based on color or texture similarity. The segmentation of a scene is shown with the results in section 5.

3.2.1 Super-pixels

We have used over-segmentation using the method called SLIC proposed by [2]. Due to the persistent cycle between scribble propagation based on flow field, the feedback from the propagated alpha map of the current frame and the morphological removal of excess and overlapping scribbles, the number of foreground and background scribbles are bound to dwindle. To counter this without any human intervention, the algorithm needs to be able to access specific local regions and generate appropriate, artificial scribbles wherever necessary. For this purpose, we consider the *matte extracted background* image of every frame and over-segment it into super-pixels, as seen in results(figure 5). This yields two advantages. Firstly, it enables the algorithm to isolate and access every super-pixel region with similar texture and intensity characteristics locally and perform independent processing in each of those regions whenever necessary. Secondly, the segmentation into super-pixels and the eventual classification into definite foreground region, definite background region, and hybrid region reduces the chances of obtaining insufficient or worse inaccurate scribbles in critical areas of the scene, which may result in the misclassification error propagating through feedback. Hence, the integration of super-pixels allows for a cleaner matte with lesser artifacts and also helps to confidently distinguish between foreground and background in critical regions. This avoids error propagation in complex scenes. Another motivation behind using super-pixels is that, by changing the parameters of the over-segmentation, the algorithm can easily be customized to obtain better and more accurate results which enhances the performance.

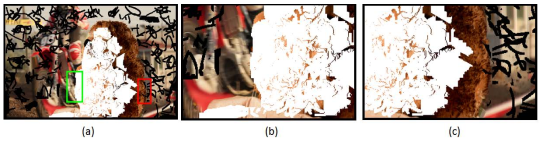

Figure 2: (a) Scribbled 112, (b) Zoomed Overflowing foreground(Green box), and (c) Zoomed penetrating background (Red box).

3.2.2 Scribbles using Bezier curves

In the proposed approach, Bezier curves are used as a medium to simulate the actual scribbles drawn by a human. This step follows the over-segmentation of the *matte extracted background* image in the super-pixel over-segmentation step. These curves are carefully drawn in selected regions where deficiency of natural scribbles leads to degradation of the matte quality or leads to the foreground-background misclassification error to be propagated. As these computer graphics tools are used as artificial scribbles, their shape and curvature are chosen to be sufficient enough to include maximum variations of scene colors as definite foreground or definite background pixels. This ensures that our algorithm generates sufficient scribbles on its own in those regions where the identity of the region is determined with confidence as either foreground or background, but there are little to no scribbles to indicate the same to the closed form scribble-based matting used on every frame. Hence, this ensures high scribbled-area-to-total-area ratio on the propagated scribbled image. A good quality matte with faster computation, without any human intervention or key frame editing can be obtained.

4. IMPLEMENTATION

4.1 The Propagation

The extracted background image is segmented into 50-100 super-pixels. Each super-pixel region obtained is classified into definite foreground super-pixels, definite background super-pixels and hybrid super-pixels based on the count of pixels which are extracted from the image as foreground. The following operation on every definite super-pixel(background or foreground) obtained is then performed.

If the current super-pixel belongs to background, then number of points is chosen to be 6 to create a Bezier curve. Within the current super-pixel, the first point is chosen randomly. Then we take the mirror image of first point with respect to the centroid of the current super-pixel to fix the second point out of the 6 points. To get the remaining points, we rotate and translate the position vector of the second point by a random angle and a random magnitude for a total of $(n-2)$ times. The Bezier curve is interpolated through these n points and all the points lying on the created curve are appended to the array *far-background* scribbles, containing the naturally propagated background scribbles. If the current super-pixel belongs to foreground, similar iterative operation is performed to choose the optimal set of points and the resulting points lying on the Bezier curve are included in an array, *far-foreground*, containing the naturally propagated foreground scribbles.

Next, super-pixels belonging to the hybrid regions are considered. Such super-pixels typically belong to the regions close to the boundary separating the foreground and background, and hence require careful designation of appropriate scribble points. More number of accurate scribbles generated at these regions aid in getting high quality mattes in complex scenes.

Average foreground optical flow of the current hybrid super-pixel region is calculated, by keeping track of any *far-foreground* pixels available in this super-pixel region. Such points act as reference for computing the average. If no *far-foreground* pixel is found or if there are lesser number of *far-foreground* pixels than a threshold T_f, average through the dark pixels of *matte extracted background* image (extracted foreground pixels) is calculated. That is, the pixels which are below a particular tri-channel threshold $[T_1, T_2, T_3]$ in the RGB space.

To calculate the average background optical flow of the hybrid super-pixel, the points corresponding to the *far-background* scribbles include which are inside the nearest definite background super-pixel as measured from the centroid of the current hybrid super-pixel. *Near foreground* and *near background* points are carefully deduced, by comparing their flow values against a weighted difference of average background optical flow and average foreground optical flow. There will be two thresholds (γ_r, γ_c) for detecting *near foreground* points and two other similar thresholds (δ_r, δ_c) of weights for detecting *near background* points. This operation is performed until all the hybrid super-pixel regions obtained are exhausted.

Finally, these new *near foreground* and *near background* pixels are appended to the array *far foreground* scribbles and *far background* scribbles respectively. At the end of this process, the complete collection of both naturally propagated and artificially generated background and foreground scribble points are available. These points are combined with appropriate color in the next frame of the dataset.

4.2 Performance parameters

With lesser amount of intensity change in the neighbourhood of different region pixels, the parameters for the natural image matting can be set to lower values. For the most of the scenes taken for our dataset, it is *level number = 2* and *active level number = 2* for natural image matting. The images given as input, originally of size 1080×1920, are resized to get images of size 388×584 for faster computations. For the first frame of the sequence, manually generated scribbled image is given to the algorithm after experimenting with scribble amount and algorithm parameters to get the best initial foreground matte.

Table 1: Performance Parameters for datasets

S.No.	Sequence	super-pixels (k)	(γ_r, γ_c)	(δ_r, δ_c)
1	*Dmitriy*	100	[1,1]	[2.5,2]
2	*City*	50	[1,1]	[2.5,2]
3	*Flowers*	70	[1,1]	[5.5,5.5]
4	*Snow*	100	[1,1]	[2.5,2]
5	*Alex*	100	[0.3,0.3]	[3.5,3]
6	*Slava*	100	[1,1]	[2.5,2]

Besides using the parameters (default being $(\gamma_r, \gamma_c) = [1, 1]$ & $(\delta_r, \delta_c) = [2.5, 2]$) described in the given methodology, other parameters control the algorithm robustness such as the number of super-pixels (default: $k = 100$) that are used in different datasets. There also occurred the need to change the optical flow difference thresholds with the datasets because of change in relative motion between foreground and background. The respective parameters are shown in table 1.

5. RESULTS

5.1 Discussion

The proposed algorithm was executed on MATLAB 2012a, on a Linux system with Intel Core i7 and 16 GB RAM. The entire computation of the algorithm took on an average of 16 seconds/frame for the dataset from [13]. The images of results from different videos are displayed in figure 5. As seen in the first row, frame 15 from the *Dmitriy* sequence consists of a complex background scene, several kinds of movement for the foreground teddy bear and also movement of the capture camera. The proposed algorithm is successfully able to generate sufficient accurate scribbles at precise locations, in order to generate a good quality matte for a frame in the middle of the sequence. The second row shows frame 17 of the same sequence where the teddy bear rotates slightly to its left and new *far foreground* and *near foreground* scribes are propagated and generated, albeit a lack of background scribes in the area right of the teddy bear is seen. However, the proposed algorithm is able to detect that and is able to recover in frame 19 by generating sufficient background scribbles in the scribble-deficient areas, generating a significantly better matte than that of for frame 17. The fourth row shows the continual performance of the proposed algorithm throughout the full sequence at frames 2, 57, 101, and 146, the performance and quality remains quite stable throughout the entire range of motion of the complex scene.

The fifth row shows the performance of the proposed algorithm on the *Flowers* sequence. We can notice the precise labelling of background pixels in the tapering area right of

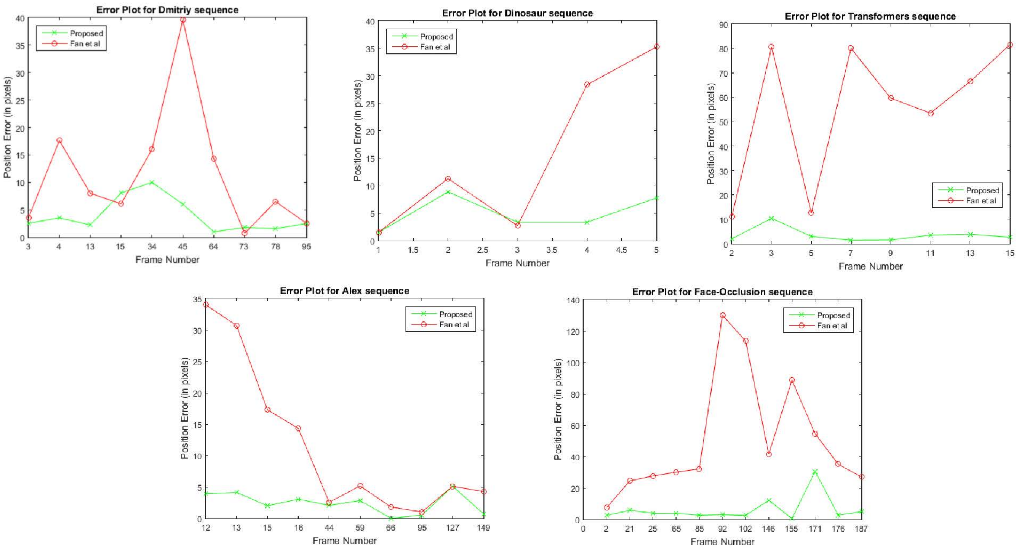

Figure 3: Quantitative Comparison: Error plots of different test sequences[1] with respect to Scribble Tracker[14].

Table 2: Error rate comparison

Dataset	min rate (Ours)	avg rate (Ours)	avg rate (Closed-form)
City	6.0994	8.8737	0.1507
Snow	1.7672	3.9894	0.8724
Slava	3.4731	5.5881	0.1335

the lady, in spite of a different, brownish color pattern which is a newly exposed color sample in the entire background. The sixth row shows our results on the *City* sequence. We can notice how the car in the initial frame passes from left to right through the face, and is partially occluded, whereas it has completely passed the face and is fully exposed after being occluded for quite a few frames in the right frame. Still, the proposed approach yields good quality mattes in both the cases. The seventh row shows the results on the *Snow* sequence, where the right frame has the lady lifting her arm, introducing an unseen part of the foreground object of interest, yet our algorithm is able to distinguish the newly introduced arm as part of the foreground. Also, we can notice how the algorithm is able to see through the near similarity in colors between the foreground and background, and nonetheless able to deliver good results.

5.2 Qualitative and Quantitative Comparisons

Quantitative comparisons have been made, one of which is shown in figure 3. The proposed method significantly outperforms the other method of similar approach to the problem (refer [14] for details on how the comparison has been made). Another comparison is made in table 2 using an error rate $ER = mean\left(\frac{\alpha_x(i+1)-\alpha_x(i)}{I_x(i+1)-I_x(i)}\right), \forall(x,i)$, where x is the pixel index in our image [30]. i and $i+1$ denote the current and following frame in the sequence. α is the generated alpha masks of original frames I. Qualitative comparisons have also been made on the dataset *city* with KNN and Snapcut [5, 8]. As seen in figure 4, our method can generate a crisper matte and proves to be robust to illumination changes.

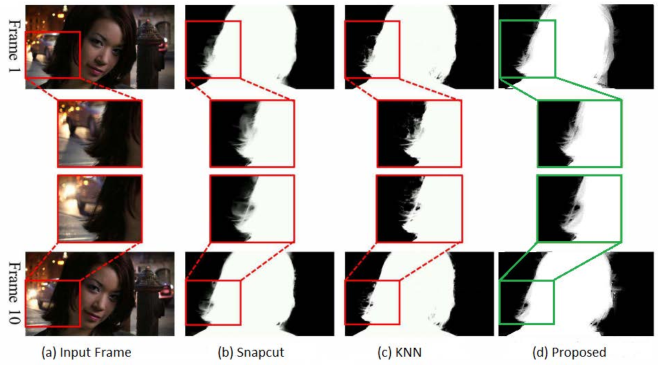

Figure 4: Comparisons on the sequence *City* with Snapcut [5] and KNN [8].

We can see from table 2 that our method lacks accuracy compared to the ground truth. The ground truth is extracted by manually specifying the trimap for every frame using closed form solution for image matting. On the other hand, our method is fully automatic and makes use of scribbles. (Refer supplementary material.)

5.3 Limitation

In the *Slava* sequence, where there was extremely subtle relative motion between the background and foreground in certain select hybrid super-pixel regions, misclassification of foreground and background points was observed. Although the algorithm recovered immediately in the next frame as per the design, but the occurrence of misclassification was significantly more frequent compared to the other scenario. While the *City* sequence also faced the same problem with the face moving extremely slowly, there were sufficient cues in the relatively more active background for the algorithm

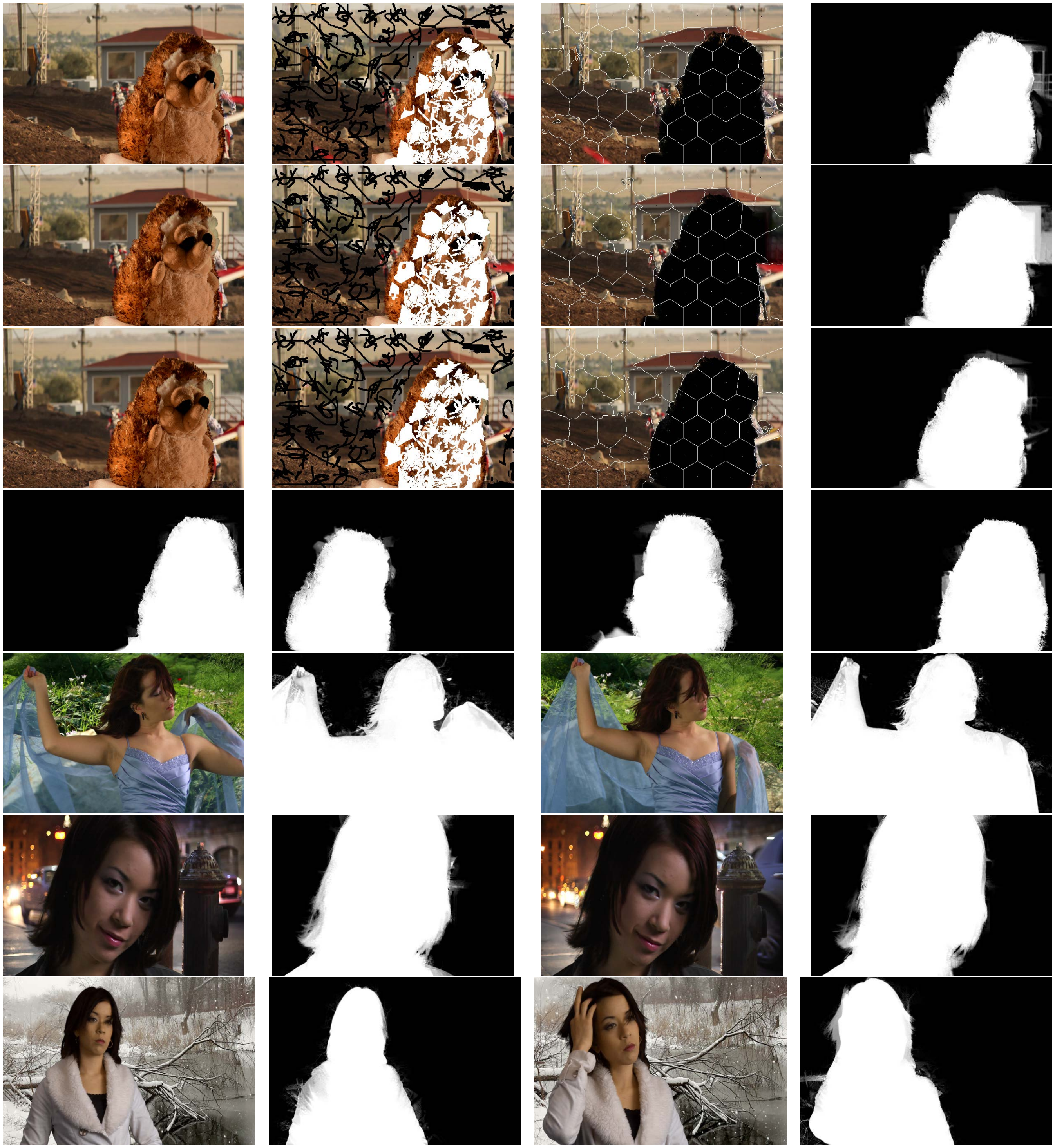

Figure 5: **First row**: *Dmitriy* sequence, Frame 15, Propagated scribes, Super-pixel over-segmentation of the matte extracted background frame and the corresponding alpha mask ; **Second row**: *Dmitriy* sequence, Frame 17, Propagated scribes, Super-pixel over-segmentation of the matte extracted background frame and the corresponding alpha mask ; **Third row**: *Dmitriy* sequence, Frame 19, Propagated scribes, Super-pixel over-segmentation of the matte extracted background frame and the corresponding alpha mask ; **Fourth row**: *Dmitriy* sequence, Frames 2, 57, 101, 146; **Fifth row**: *Flowers* sequence, Frames 3 and 92 with their respective alpha masks; **Sixth row**: *City* sequence, Frames 8 and 72 with their respective alpha masks; **Seventh row**: *Snow* sequence, Frames 31 and 119 with their respective alpha masks.

to correctly distinguish between the two regions and avoid misclassification.

6. CONCLUSION

We have introduced a new approach towards scribble-based natural video matting based on optical flow estimation. The proposed solution has been verified on several standard sequences, with each sequence depicting a different challenging dynamic scene. We have shown that, similar to discrimination between color samples, optical flow is good as a criterion for distinguishing foreground and background precisely enough to generate new scribbles for obtaining high quality mattes. The proposed algorithm uses user involvement for only the first frame to initialise the scribbles and the propagation is carried out without the necessity of new set of user scribbles. The method can be improved using optimization methods, implemented on super-pixels corresponding to hybrid regions, for higher accuracy mattes. Further, we shall investigate the duration of the video sequence upto which one can generate mattes without user interference.

7. ACKNOWLEDGEMENT

The authors would like to thank SERB-DST for the financial support through start-up grant for young scientists.

8. REFERENCES

[1] Tracking dataset. http://cmp.felk.cvut.cz/˜vojirtom/dataset/. Accessed: 2016-10-12.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. Technical report, 2010.

[3] N. Apostoloff and A. Fitzgibbon. Bayesian video matting using learnt image priors. In *CVPR*. IEEE, 2004.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*. IEEE, 2009.

[5] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics (TOG)*, volume 28, page 70. ACM, 2009.

[6] A. Berman, A. Dadourian, and P. Vlahos. Method for removing from an image the background surrounding a selected object, Oct. 17 2000. US Patent 6,134,346.

[7] A. Berman, P. Vlahos, and A. Dadourian. Comprehensive method for removing from an image the background surrounding a selected subject, Oct. 17 2000. US Patent 6,134,345.

[8] Q. Chen, D. Li, and C.-K. Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013.

[9] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski. Video matting of complex scenes. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 243–248. ACM, 2002.

[10] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *CVPR*. IEEE, 2001.

[11] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631–1643, 2005.

[12] P. Doğan, T. O. Aydın, N. Stefanoski, and A. Smolic. Key-frame based spatiotemporal scribble propagation. In *Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing*, pages 13–20. Eurographics Association, 2015.

[13] M. Erofeev, Y. Gitman, D. Vatolin, A. Fedorov, and J. Wang. Perceptually motivated benchmark for video matting. In *BMVC*, 2015.

[14] J. Fan, X. Shen, and Y. Wu. Scribble tracker: a matting-based approach for robust tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1633–1644, 2012.

[15] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*. IEEE, 2010.

[16] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM, 2004.

[17] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, 2008.

[18] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Transactions on Graphics (TOG)*, 24(3):595–600, 2005.

[19] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 567–576. ACM, 2005.

[20] Y. Mishima. Soft edge chroma-key generation based upon hexoctahedral color space, Oct. 11 1994. US Patent 5,355,174.

[21] R. J. Qian and M. I. Sezan. Video background replacement without a blue screen. In *ICIP*. IEEE, 1999.

[22] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*. IEEE, 2007.

[23] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *CVPR*. IEEE, 2000.

[24] M. Sindeev, A. Konushin, and C. Rother. Alpha-flow for video matting. In *ACCV*, pages 438–452. Springer, 2012.

[25] A. R. Smith and J. F. Blinn. Blue screen matting. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 259–268. ACM, 1996.

[26] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*. IEEE, 2010.

[27] Z. Tang, Z. Miao, Y. Wan, and D. Zhang. Video matting via opacity propagation. *The Visual Computer*, 28(1):47–61, 2012.

[28] J. Xiao and M. Shah. Accurate motion layer segmentation and matting. In *CVPR*. IEEE, 2005.

[29] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(7):1195–1209, 2009.

[30] D. Zou, X. Chen, G. Cao, and X. Wang. Video matting via sparse and low-rank representation. In *ICCV*, 2015.